In the Claims:

Please amend claims 1, 11, and 15 as follows:

1. (currently amended) A method for implementing atomic data tracing in a processor system including an auxiliary processor unit coupled to a central processor unit (CPU), using the auxiliary processor unit (APU) to perform the steps of:

identifying a trace instruction; said trace instruction including a primary op code and indicating General Purpose Registers (GPRs) containing information to identify a first GPR containing data to be written into a current trace entry of a single trace buffer and to identify a set of trace engine registers defining a trace engine to use for said trace instruction; said trace engine including said single trace buffer; said data to be written being saved automatically in at least one GPR including said first GPR during normal context switch processing;

said trace buffer and said set of trace engine registers defining said trace engine being accessible by the APU;

signaling the CPU with a pipeline stall signal for stalling a CPU instruction stream pipeline;

checking for an enabled trace engine for said trace instruction,

writing trace data into a said single trace buffer responsive to an identified enabled trace engine for said trace instruction utilizing said set of trace engine registers defining said trace engine including writing trace data into said single trace buffer from multiple execution contexts; and said trace engine including a set of device control registers (DCRs) accessible by the APU and used to determine where to write the data

into said trace buffer; said set of device control registers (DCRs) including a trace buffer pointer register storing a base address of said trace buffer and an offset indicating a current trace buffer entry, and a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer; and

signaling the CPU with an op done signal for allowing the CPU to continue with instruction stream pipeline processing.

Claims 2-3. (canceled)

4. (previously presented) The method for implementing atomic data tracing as recited in claim 1 wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register.

5. (canceled)

6. (previously presented) The method for implementing atomic data tracing as recited in claim 1 wherein said set of device control registers (DCRs) include a control register storing an enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly.

7. (previously presented) The method for implementing atomic data tracing as recited in claim 6 wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and wherein the step of writing trace data

into said trace buffer includes writing a time stamp with said trace data responsive to said control register time stamp value.

8. (previously presented) The method for implementing atomic data tracing as recited in claim 6 wherein said control register includes a number field indicating a number of bytes to be traced; and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes.

9. (previously presented) The method for implementing atomic data tracing as recited in claim 1 wherein said trace instruction includes a number field indicating a number of bytes to be traced; and wherein the step of writing trace data into said trace buffer includes updating said offset into said current trace buffer entry of said trace buffer pointer register by said number of bytes.

10. (previously presented) The method for implementing atomic data tracing as recited in claim 1 responsive to identifying no enabled trace engine for said trace instruction, signaling the CPU with said op done signal for allowing the CPU to continue with instruction processing without writing trace data.

11. (currently amended) An apparatus for implementing atomic data tracing in a processor system including an auxiliary processor unit (APU) coupled to a central processor unit (CPU), said apparatus comprising:

a trace engine; said trace engine including a set of device control registers (DCRs) accessible by the APU, and a single trace buffer; said set of device control registers (DCRs) including a trace buffer pointer register storing a base address of said

trace buffer and an offset indicating a current trace buffer entry and including a base address mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer;

a trace instruction; said trace instruction including a primary op code and encoded first and second general purpose registers (GPRs), said first GPR containing an index to said trace engine DCRs and said second GPR indicating a first GPR containing data to be written into a current trace entry in said trace buffer; said data to be written being saved automatically in at least one GPR including said first GPR during normal context switch processing;

the APU processes said trace instruction performing the steps of signaling the CPU with a pipeline stall signal for stalling a CPU instruction stream pipeline; responsive to identifying an enabled trace engine for said trace instruction, writing trace data into said single trace buffer utilizing said set of device control registers (DCRs) included in said trace engine to determine where to write the data into said single trace buffer and including writing trace data into said single trace buffer from multiple execution contexts; and signaling the CPU with an op done signal for allowing the CPU to continue with instruction stream pipeline processing.

12. (previously presented) The apparatus for implementing atomic data tracing in a processor system as recited in claim 11 wherein said set of device control registers (DCRs) include a control register storing an enabled bit indicating whether or

not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly.

13. (previously presented) The apparatus for implementing atomic data tracing in a processor system as recited in claim 12 wherein the APU updates said offset to said current trace buffer entry of said trace buffer pointer register for each trace data entry written to said trace buffer.

14. (previously presented) The apparatus for implementing atomic data tracing in a processor system as recited in claim 12 wherein said control register includes a time stamp value indicating whether or not a time stamp should be traced; and the APU writes a time stamp with said trace data responsive to said control register time stamp value.

15. (currently amended) A computer program product for implementing atomic data tracing in a processor system including an auxiliary processor unit (APU) coupled to a central processor unit (CPU), said computer program product including instructions stored on a computer recording medium consisting one of a floppy disk, an optically read compact disk, a compact disk read only memory (CD-ROM), and a tape, wherein said instructions, when executed by the processor system to cause the processor system to perform the steps of:

defining a single trace buffer and a set of device control registers (DCRs) accessible by the APU to create a trace engine; said set of device control registers (DCRs) including a trace buffer pointer register for storing a base address of said trace buffer and an offset indicating a current trace buffer entry and including a base address

mask register storing a mask indicating which bits in said trace buffer pointer register hold said base address and which hold said offset; said base address mask register used to determine a wrap point of said trace buffer;

providing a trace instruction; said trace instruction including a primary op code and encoded first and second general purpose registers (GPRs), said first GPR containing an index to said trace engine DCRs and said second GPR indicating a first GPR containing data to be written into a current trace entry in said trace buffer; said data to be written being saved automatically in at least one GPR including said first GPR during normal context switch processing; and

utilizing the APU for processing said trace instruction by performing the steps of: signaling the CPU with a pipeline stall signal for stalling a CPU instruction stream pipeline; utilizing said set of DCRs, checking for an enabled trace engine for said trace instruction; and responsive to identifying an enabled trace engine for said trace instruction, writing trace data into said single trace buffer utilizing said set of device control registers (DCRs) included in said trace engine to determine where to write the data into said trace buffer including writing trace data into said single trace buffer from multiple execution contexts; and signaling the CPU with an op done signal for allowing the CPU to continue with instruction stream pipeline processing.

16. (previously presented) The computer program product for implementing atomic data tracing as recited in claim 15 wherein the step of defining a set of device control registers (DCRs) accessible by the APU to create a trace engine includes the steps of defining said set of DCRs including a control register storing an

enabled bit indicating whether or not said trace engine for said trace instruction is enabled, said enabled bit being used for allowing data tracing to be turned on and off on the fly.

17. (previously presented) The computer program product for implementing atomic data tracing as recited in claim 16 wherein the step of writing trace data into said trace buffer includes updating said offset to said current trace buffer entry of said trace buffer pointer register for each trace data entry written to said trace buffer.

18. (previously presented) The computer program product for implementing atomic data tracing as recited in claim 16 wherein the step of writing trace data into said trace buffer includes checking a time stamp value in said control register indicating whether or not a time stamp should be traced; and the APU writes a time stamp with said trace data responsive to said control register time stamp value.